Министерство образования и науки Российской Федерации

Новосибирский государственный технический университет

Кафедра теоретической и прикладной информатики

Индивидуальная работа

по дисциплине

«Управление ресурсами в вычислительных системах»

УПРАВЛЕНИЕ РЕСУРСАМИ В ОС WINDOWS

Факультет: ПМИ

Группа: ПМИ-12

Студент: Швадченко А. В.

Вариант: 20

Преподаватели: Стасышин В. М.

Новосибирск

2024

1. ***Цель работы и задача***

Разработать графическое приложение с использованием WinAPI для определения дня недели и размера кэша данных второго уровня, для практического освоения вопросов управления ресурсами ОС Windows. Изучить основные особенности ОС Windows: работа с потоками, организация графического интерфейса, механизм обработки сообщений, использование динамически подключаемых библиотек (dll), низкоуровневое взаимодействие с процессором.

1. ***Метод решения***

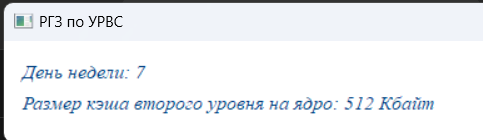
День недели можно определить с помощью структуры SYSTEMTIME, используя wDayOfWeek. В результате будет возвращено значение от 0 до 6, где 0 – воскресенье, 1-понедельник, 2- вторник и т.д.

Размер кэша можно определить, используя инструкции процессора CPUID. Для процессоров Intel и AMD достаточно вызвать CPUID c входным значением EAX = 0x80000006, в результате размер кэша L2 находится в ECX[31:16].

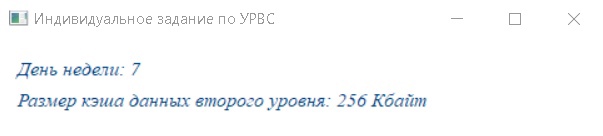
1. ***Спецификация***

Программа находится в файле main.cpp, динамическая библиотека в info.cpp. Для получения .exe файла необходимо запустить файл compile.bat в командной строке разработчика Visual Studio.

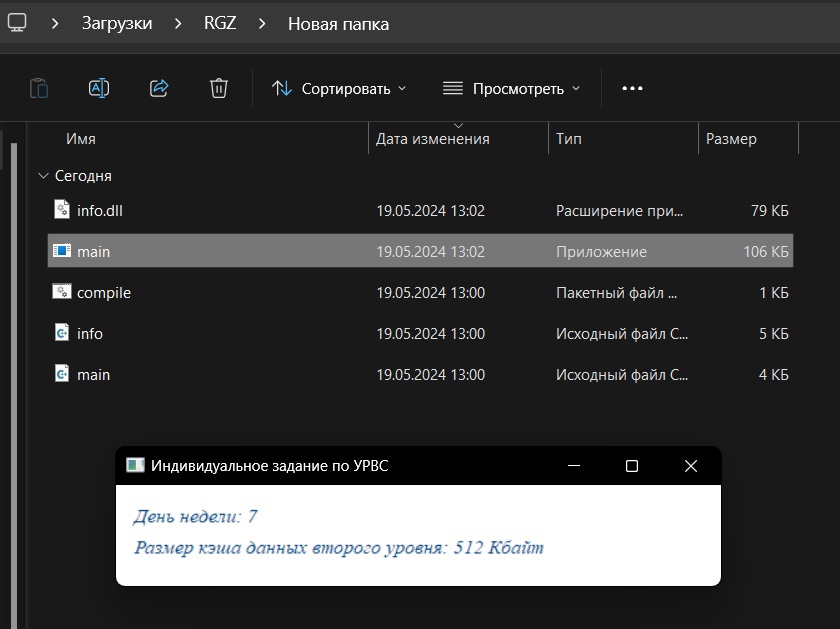
1. ***Тесты***
2. AMD Athlon Gold 3150U ядра: 2, кэш L2: 1мб



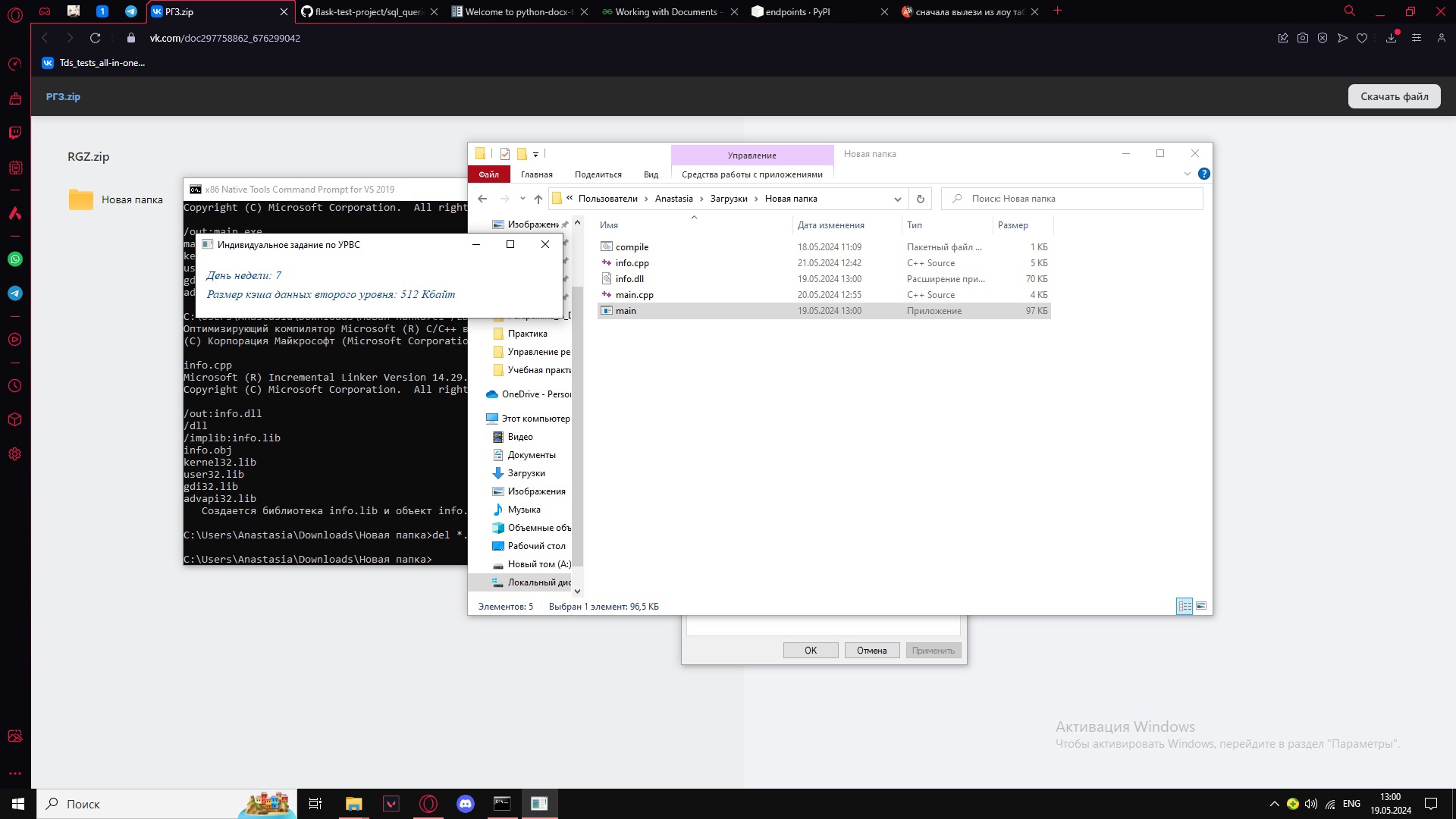
1. Intel Core i5-9300H ядра: 4, кэш L2: 1мб



1. AMD Ryzen 7 5800U ядра: 8, кэш L2: 4 мб



1. AMD Ryzen 7 4800H ядра: 8, кэш L2: 4 мб



1. ***Текст программы***

***Библиотека info.cpp***

#define \_CRT\_SECURE\_NO\_WARNINGS

#define UNICODE

#include <stdio.h>

#include <tchar.h>

#include <windows.h>

using namespace std;

extern "C" \_declspec(dllexport) int Day() //функция для определения дня недели

{

SYSTEMTIME systemTime;

GetSystemTime(&systemTime);

return systemTime.wDayOfWeek;

}

extern "C" \_declspec(dllexport) int L2size()//функция для определения размера кэша 2 уровня

{

int iebx;

\_asm

{

mov eax, 0x80000006

cpuid

push ecx

xchg eax, ecx

and eax, 0xff;

pop ebx

shr ebx, 16;

mov iebx, ebx

}

return(iebx);

}

***Основная программа main.cpp***

#define \_CRT\_SECURE\_NO\_WARNINGS

#define UNICODE

#include <windows.h>

#include <tchar.h>

#include <string>

#define BUFFER\_SIZE 64

LPCWSTR szClassName = L"MyClass";

LPCWSTR szTitle = L"РГЗ по УРВС";

TCHAR DAY [BUFFER\_SIZE];

DWORD DAYlen;

DWORD L2Size;

TCHAR L2Info[BUFFER\_SIZE];

DWORD WINAPI ThreadFunc(void\*)

{

// Подключение библиотеки

typedef int(\*ImportFunction)(char\*);

ImportFunction D;

ImportFunction L2;

HINSTANCE hinstLib = LoadLibrary(TEXT("info.dll"));

D = (ImportFunction)GetProcAddress(hinstLib, "Day");

L2 = (ImportFunction)GetProcAddress(hinstLib, "L2size");

char String[BUFFER\_SIZE + 8];

DAYlen = BUFFER\_SIZE - 16;

sprintf(String, "%i", DAYlen);

// Вызов функций из динамической библиотеки

int day = D("");

int l2 = L2("");

/\* Освобождение дескриптора \*/

FreeLibrary(hinstLib);

if (day == 0)

day = 7;

wsprintf(DAY, L"День недели: %i", day);

wsprintf(L2Info, L"Размер кэша второго уровня на ядро: %i Кбайт", l2);

return 1;

}

LRESULT CALLBACK WindowFunc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam)

{

PAINTSTRUCT ps;

HDC hDC;

HFONT hFont = CreateFont(18, 0, 0, 0, FW\_THIN, TRUE, FALSE, FALSE, DEFAULT\_CHARSET, OUT\_OUTLINE\_PRECIS,

CLIP\_DEFAULT\_PRECIS, CLEARTYPE\_QUALITY, VARIABLE\_PITCH, TEXT("Times New Roman"));

switch (msg)

{

case WM\_CREATE:

// Создание потока

HANDLE hThread;

DWORD IDThread;

hThread = CreateThread(NULL, 0, ThreadFunc, NULL, 0, &IDThread);

WaitForSingleObject(hThread, INFINITE);

CloseHandle(hThread);

break;

case WM\_DESTROY:

// Закрытие окна

PostQuitMessage(0);

break;

case WM\_PAINT:

// Инициализация контекста устройства

hDC = BeginPaint(hWnd, &ps);

SelectObject(hDC, hFont);

SetTextColor(hDC, RGB(0, 64, 128));

TextOut(hDC, 15, 17, DAY, (int)DAYlen + 16);

TextOut(hDC, 15, 42, L2Info, 50);

EndPaint(hWnd, &ps);

break;

default:

return DefWindowProc(hWnd, msg, wParam, lParam);

}

return 0;

}

int WINAPI WinMain(HINSTANCE hThisInst, HINSTANCE hPrevInst,

LPSTR str, int nWinMode)

{

MSG msg;

WNDCLASS wcl;

HWND hWnd;

// Создание класса окна

wcl.hInstance = hThisInst;

wcl.lpszClassName = szClassName;

wcl.lpfnWndProc = WindowFunc;

wcl.style = CS\_HREDRAW | CS\_VREDRAW;

wcl.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wcl.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcl.lpszMenuName = NULL;

wcl.cbClsExtra = 0;

wcl.cbWndExtra = 0;

wcl.hbrBackground = (HBRUSH)GetStockObject(DEFAULT\_PALETTE);

RegisterClass(&wcl);

// Создание окна на базе созданного класса

hWnd = CreateWindow(szClassName, szTitle, WS\_OVERLAPPEDWINDOW |

WS\_CLIPCHILDREN | WS\_CLIPSIBLINGS, 100, 50, 700, 120,

HWND\_DESKTOP, NULL, hThisInst, NULL);

// Отображение окна

ShowWindow(hWnd, nWinMode);

UpdateWindow(hWnd);

//Цикл обработки сообщений

while (GetMessage(&msg, NULL, 0, 0)) // Получение сообщения

{

TranslateMessage(&msg); // Преобразование виртуальных кодов клавиш в ASCII-значения

DispatchMessage(&msg); // Посылка сообщения в нужную оконную процедуру

}

return msg.wParam;

}